GWD-I
Category: Informational
Preservation Environments Research Group

Bruce Barkstrom, NASA Langley Research Center

12-June-2005
2-September-2005

Status of This Memo: Informational

This memo provides information to the Grid community specifying the requirements for systems that support the long-term archiving of digital material. In particular, data grids provide the capabilities that address each of the archiving requirements.

Copyright Notice

Long-Term Digital Archive Requirements

Abstract

The core requirements for long-term digital archives can be expressed as management policies for both the digital records and the infrastructure that supports the digital records. Sixteen core requirements have been identified. This document explains the significance of each core requirement, and proposes multiple levels of support for each core requirement.

Contents

1. Preservation in Archives

Archives exist to maintain a long-term record of their contents. In practical terms, this means preserving the records in the archive for at least 200 years. While much of the archival community's attention has been focused on dealing with the technological issues involved in this preservation requirement, it seems likely that the more difficult issues are those that arise from sociological and resource issues. Indeed, it may be that the most difficult issue is preserving a community that understands the contents and uses of the archived material.

2. Fundamental Requirements for Long-Term Digital Archives

A long-term digital archive provides support for authenticity (the assurance that the material in the digital archive is correctly linked to descriptions of its origin), integrity (the assurance that the material in the archive is uncorrupted, that the chain of custody can be tracked, and that the information content remains unchanged), and infrastructure independence (the assurance that the digital archive has not imposed any proprietary standards that prevent migration of the contents of the digital archive to another choice of technology).

Use of data in an archive can be rendered impossible by at least five risk factors:

1. Malicious or inadvertent destruction
2. Technological Obsolescence (hardware or software)
3. Loss of context (undocumented features; loss of understanding; aging, retirement, or death of key community members)
4. Competition for resources with new opportunities
5. Institutional instability

These factors impact two features of long-term archives in very important ways. They create two fundamental requirements:

1. A requirement to reduce operation costs as much as possible
2. A requirement to minimize errors in all of the archive operations.

3. Level 1 Derived Requirements

3.1 Core Requirements for Long-Term Digital Archives

There are sixteen Core Requirements that derive directly from the Fundamental Requirements for a long-term digital archive:

1. Unless the probability of loss per year from intrusion is less than 0.00005, a long-term archive will require off-line (and off-site) storage.
2. The data provider and the archive must agree on a valuation and risk assessment for the data and metadata to be stored in the archive.
3. The archive needs to provide redundant systems to avoid single point of failure modes that would lead to loss of data, and mechanisms to validate the consistency of the data across the redundant systems.
4. An archive system needs to be independent of the language in which it is implemented and of the operating system on which it runs. This derived requirement is necessary in order to ensure that the archive has independent modes of failure arising from system design errors and errors in system implementations. Thus, if possible, a long-term

archive should be independently implemented in at least two languages, and preferably three or more.

5. A long-term archive cannot rely on hardware provided by a single vendor.
6. A long-term archive cannot rely on software provided by a single vendor when implemented using proprietary software.
7. The archive needs to document its contents, the software components of its systems, as well as the procedures it uses to ingest, store, and distribute data.
8. An archive needs a method of registering permanent names for the unique content in its collections, for the archivists running the system, and for the processes used to manage the content.
9. Because metadata may “reach inside” files, an archive needs a method of registering permanent names for data elements inside files, independent of possible permutative rearrangement of either the files or their format.
10. The archive needs to make systematic plans for preserving and evolving the intellectual capital represented by its data, metadata, and documentation through the changes required by the technological and sociological evolution of the archive and its environment.
11. The archive needs to actively create a dispersed community of practice and discourse that is familiar with the contents and procedures of the archive. This requirement suggests that wherever possible, the archive should consider the way in which it can foster federations with other archives (to reduce the probability of loss by dispersing the archive’s contents) and a vibrant Open Source community.
12. A long-term archive needs to actively work to automate as much of its operations as possible. General experience suggests that over long time periods, costs of human activity are the largest element in the cost of operations.
13. Specific experience in the NASA ESE data centers also suggests that automation must be designed into the archive’s systems, rather than added to them later.
14. An archive needs a rigorous cost model for its operational costs.
15. The archive cost model needs to use statistical information from the actual history of the archive to project future costs where this history is available. In other words, the archive systems must be designed to collect both a record of archive activities and archive costs.
16. An archive needs to develop an Open Source archival community that can accept stewardship for preservation of the intellectual capital contained in archives.

4. Factors That Create Difficulties for Long-Term Digital Archives

4.1 Malicious or Inadvertent Destruction

Under normal archival standards, data needs to have a high probability of surviving more than 200 years. If the archive is connected to the Internet, it appears that a reasonable probability for an intrusion with potential consequences for the data, the metadata, or the chain of custody for both is about 10% per year. This rather high risk means that without strenuous effort on the part of the archive, the probability of having data survive 200 years is (1-0.1)**200 or about 7x10**(-10). These are clearly rather stiff odds. At present the number of intrusion incidents is rising. At the same time, an archive can take a number of steps to decrease the probability of loss – most notably by sending data to a site away from the archive. Such off-site storage, in which the data and metadata are not connected to the network, appears to be a very important component in reducing the risk of data loss.

It is also important to note that the quantification of the risk we have just identified must be considered in the light of the cost of replacing the data and metadata. Technically, this cost is part of the valuation of the data and metadata – and, in this case, is based on replacement costs. If the loss of the data is of no particular consequence, then we might not consider it worth-while to undergo the expense and hassle of off-site storage. To be careful, we need to ensure that the archive and the data provider have carefully evaluated the probability of loss – and the value of the replacement. In other words, the archive and the data provider must quantify the probability of risk and the cost of preservation.

This pessimistic assessment leads to two derived requirements:

1. Unless the probability of loss per year from intrusion is less than 0.00005, a long-term archive will require off-line (and off-site) storage.
2. The data provider and the archive must agree on a valuation and risk assessment for the data and metadata to be stored in the archive.

4.2 Technological Obsolescence (Hardware or Software)

Unfortunately, data are not safe – even when we put it on tapes and store it deep under a mountain. Every five years or so (at least for the foreseeable future), the vendors of the hardware used to read the storage media (tapes or CDs or holographic media …) produce new models of the devices that read or write the data onto the storage media. Likewise, the vendors of software produce new versions of their wares on a time scale of about eighteen months. These facts of life mean that we must expect to move data in storage (or in the archive itself) from one medium to another about once every five years. This means that the transfer process will only occur about forty times in two hundred years – not two hundred times. While such transfers appear to be reasonably safe, experience in ASDC (and in other archives) suggests that the probability of successful transfers from one medium to another has a definitely finite probability of loss, which we estimate at perhaps 2% per transfer. This numerical value is derived by considering our experience with both large-scale data transfers and routine operations within ASDC. For example, in one recent incident, a router failure corrupted about 10% of the data in a rather large dataset delivered from another data center. This problem has happened relatively rarely, but such incidents appear to happen about once every five years or so.

Quantitatively, with a 2% probability of loss per transfer, the probability of having a particular file survive 200 years is (1 – 0.02)**40 or about 0.45. This probability is much higher than that for loss through malicious or inadvertent destruction. At the same time it is unacceptably low for real archival work, where the probability for survival in each data migration needs to be well above 0.99. Translated into an allowable probability of loss per transfer (and assuming a transfer will need to occur once every five years), this means that the allowable probability of loss per transfer needs to be kept below about 0.0002. This is rather stringent.

In terms of the data contamination incident we just described, it also means that an archive will need to play a very active role in reducing the probability of errors – which may arrive in quite unexpected forms (such as unexpected vendor hardware failures). Given the stringency of this requirement, the archive needs a carefully planned strategy to reduce the role of happenstance. A key element of such a strategy is to reduce the probability of single point failures – or to increase the redundancy of the system, or to increase the number of end-to-end validation steps used to assure integrity. In other words, if the archive can reasonably assume that data loss incidents will occur independently on separately instantiated systems, then the probability of unsustainable data loss over 200 years may be reduced to more acceptable levels. In other words, we have another set of derived requirements:

3. The archive needs to provide redundant systems to avoid single point of failure modes that would lead to loss of data, and mechanisms to validate the consistency of the data across the redundant systems.
4. An archive system needs to be independent of the language in which it is implemented and of the operating system on which it runs. This derived requirement is necessary in order to ensure that the archive has independent modes of failure arising from system design errors and errors in system implementations. Thus, if possible, a long-term archive should be independently implemented in at least two languages, and preferably three or more.
5. A long-term archive cannot rely on hardware provided by a single vendor.

6. A long-term archive cannot rely on software provided by a single vendor when implemented using proprietary software.

4.3 Loss of Context

While we are familiar with the corrosive impact of Moore's law on the stability of hardware and software, a long-term archive also needs to consider the longer term impact of changes in a number of "sociological" factors in its environment. These include the loss of knowledge regarding undocumented features of system software or hardware and the loss of understanding associated with aging, retirement, or death of key community members. For example, in the ASDC context, the criteria used by instrument teams to select data for calibration coefficients are probably lost once an instrument team disbands. Once this information is lost, it cannot be recovered.

While loss of context is probably a phenomenon with a longer time constant than either loss by malicious activities or technological obsolescence, an archive must still actively work to prevent it from causing additional data loss. This leads to at least five additional derived requirements:

7. The archive needs to document its contents, the software components of its systems, as well as the procedures it uses to ingest, store, and distribute data.
8. An archive needs a method of registering permanent names for the unique content in its collections, for the archivists running the system, and for the processes used to manage the content.
9. Because metadata may "reach inside" files, an archive needs a method of registering permanent names for data elements inside files, independent of possible permutative rearrangement of either the files or their format.
10. The archive needs to make systematic plans for preserving and evolving the intellectual capital represented by its data, metadata, and documentation through the changes required by the technological and sociological evolution of the archive and its environment.
11. The archive needs to actively create a dispersed community of practice and discourse that is familiar with the contents and procedures of the archive. This requirement suggests that wherever possible, the archive should consider the way in which it can foster federations with other archives (to reduce the probability of loss by dispersing the archive's contents) and a vibrant Open Source community.

4.4 Competition of Resources

While the concern over loss drives archive requirements toward increasing the investment in redundancy (both of hardware and of software), the environment in which archives operate can severely constrain the available resources. In general, archive budgets compete for resources required by organizations that see new opportunities. For example, NASA's Earth Science Enterprise (when that existed during the last decade of the Twentieth Century) faced severe competition between the requirements for operating a data and information system that had not been designed for low total cost of operation and its need to develop new missions. As another example, libraries in universities compete with other campus organizations for capital and operating budgets.

This competition leads to a new set of derived requirements:

12. A long-term archive needs to actively work to automate as much of its operations as possible. General experience suggests that over long time periods, costs of human activity are the largest element in the cost of operations.
13. Specific experience in the NASA ESE data centers also suggests that automation must be designed into the archive's systems, rather than added to them later.
14. An archive needs a rigorous cost model for its operational costs.

15. The archive cost model needs to use statistical information from the actual history of the archive to project future costs where this history is available. In other words, the archive systems must be designed to collect both a record of archive activities and archive costs.

4.5 Institutional Instability

Finally, as identified by the Library of Congress National Digital Information Infrastructure Preservation Program workshops and planning report, archives must be prepared for institutional instability. A specific example of this "instability" arises in the question of how to preserve the unique record of Earth observations created by a massive investment by NASA, although there are expectations that the long-term archival responsibility lies with NOAA. There are a number of potential loss mechanisms that may afflict the transfer of an operational archive from NASA to NOAA:

- Loss or data by name changes
- Loss of knowledge owing to perceptions that no knowledge or activity will be required to recreate higher level data products, even though science teams have spent hundreds or thousands of person hours in validating these products, which could not be reconstructed in a reasonable length of time after such a transfer
- Reduction in resources available – leading to loss of data by design, even though there are active communities still working with the data

It is not easy to deal with the issues raised by these large-scale movements of institutional resources. Perhaps the best we can do is to explore an implicit requirement:

16. Develop an Open Source archival community that can accept stewardship for preservation of the intellectual capital contained in archives.

5. Categories of Level 1 Requirements

It will be helpful to reorganize the list of Level 1 requirements we have derived from the two basic starting principles that the archive has to be prepared to avoid errors and that it needs to be as cost effective as possible. The headings that follow are organized in four categories. In some cases, we can combine Level 1 requirements. For example both 12 and 13 in the original list require system automation.

In addition, we can identify some additional requirements that we need to add to make the list more complete.

5.1 Low Total Cost of Ownership

- **Automation.** A long-term archive needs to actively work to automate as much of its operations as possible. General experience suggests that over long time periods, costs of human activity are the largest element in the cost of operations. Specific experience in the NASA ESE data centers also suggests that automation must be designed into the archive's systems, rather than added to them later. (Requirements 12 and 13)
- **Cost Model.** An archive needs a rigorous cost model for its operational costs. The archive cost model needs to use statistical information from the actual history of the archive to project future costs where this history is available. In other words, the archive systems must be designed to collect both a record of archive activities and archive costs. At the same time, the cost model must be able to incorporate new technology that may provide a substantial cost savings. (Requirements 14 and 15)
- **Commodity Computers and Data Storage.** A recent National Research Council (NRC) Report strongly recommended using commodity computers and data storage at government data centers. We concur with this recommendation. This kind of equipment lessens an archive's dependence on proprietary solutions to its problems. It also allows the equipment manufacturers to amortize their investment and increases the pool of available suppliers. Commodity data storage may increase the amount of validation that

is required for integrity checking as the systems may not be as reliable. A tradeoff analysis between labor spent on validation versus capital cost of the storage systems should be maintained for the archive.

- **Use of Open Source Software to Reduce Licensing Costs.** As with hardware, using software created and maintained by the Open Source community can substantially reduce licensing costs. The maintenance model for this software is different than that for proprietary software. Open Source software usually relies on e-mail and bug lists to maintain the software configuration. However, to the extent that we do not expect a mass market for archives, there is a requirement that the software be understood by the archival engineers. The archive will need to maintain their own testing and validation facility for the Open Source software, and may need to port their own local modifications to the Open Source software. If an archival consortium collaborates on the management of the software, the costs per institution can be minimized.

5.2 High Reliability

- **Balanced Approach to Redundancy and Dispersed Storage.** Unless the probability of loss per year from intrusion is less than 0.00005, a long-term archive will require off-line (and off-site) storage. The archive needs to provide redundant systems to avoid single point of failure modes that would lead to loss of data. (Requirements 1 and 3 with consideration of Low Total Cost of Ownership)
- **Valuation and Risk Assessment.** The data provider and the archive must agree on a valuation and risk assessment for the data and metadata to be stored in the archive. (Requirement 2)
- **Avoidance of Dependence on Proprietary Sources.** A long-term archive cannot rely on hardware provided by a single vendor. A long-term archive cannot relay on software provided by a single vendor with proprietary software. (Requirements 5 and 6)
- **Robust and Graceful Exception handling.** Neither computer hardware nor the software we develop is perfect. In addition, “things break”. In a well designed system, performance degrades gracefully, with a useful record of where the systems encountered faults and suggested fixes. In a poorly designed system, when something “breaks”, the system responds “brittlely”, with no record of where the system noted the first breakage. There are interactions between this requirement and low total cost of ownership –it’s much easier and cheaper to repair a system that identifies a fault, suggests confirmatory tests, and is able to bypass the problem until it can be tended by a help team.
- **Designed-In Security.** Security intrusions disrupt the archive’s operations and reduce its reliability. Most experts recommend that security be designed into the system “from the ground up”.

5.3 Evolvability (Infrastructure Independence)

- **Language and Implementation Independence.** An archive system needs to be independent of the language in which it is implemented and of the operating system on which it runs. This derived requirement is necessary in order to ensure that the archive has independent modes of failure arising from system design errors and errors in system implementations. Thus, if possible, a long-term archive should be independently implemented in at least two languages, and preferably three or more. (Requirement 4)
- **Modularity of Architecture with Well-Designed Message Protocols.** In a general sense, modularity of architecture requires the architect to make parts of the system that do not need to know about each other entirely separate. With good, message-passing object-oriented design, we enforce this general dictum by starting with use cases that ensure that the components of the system that do not need to “talk” with each other are ignorant and unaffected by the parts of the system that can be treated as independent. By further taking care to formalize the message protocols, we improve the modularity of the system.

- **Formalization of Operational and Evolutionary Procedures.** By formalizing the operational procedures, subjecting them to the rigor of procedural simplification (or "business process engineering"), we create a system that we can reason about. In addition, with formalization, it is more straightforward to modify the procedures when the archive needs to evolve its systems. Such modifications need to be undertaken carefully and systematically. Modularity of the architecture and formalization of procedures aid in simplifying the process.
- **Complete Documentation.** The archive needs to document its contents, the software components of its system, as well as the procedures it uses to ingest, store, and distribute data. (Requirement 7)
- **Systematic Planning for Evolution**. The archive needs to make systematic plans for preserving and evolving the intellectual capital represented by its data, metadata, and documentation through the changes required by the technological and sociological evolution of the archive and its environment. (Requirement 10)
- **Outside Participation in Design, Development, and Evolution.** In order to maximize the probability of long-term survival of knowledge, it is helpful to spread understanding of the system over a wide range of communities and to solicit their input into the system design, development, and evolution.
- **Dispersed Archive Community.** The archive needs to actively create a dispersed community of practice and discourse that is familiar with the contents and procedures of the archive. This requirement suggests that whenever possible, the archive should consider the way in which it can foster federations with other archives (to reduce the probability of loss by dispersing the archive's contents) and a vibrant Open Source community. The archive needs to develop an Open Source archival community that can accept stewardship for preservation of the intellectual capital contained in archives. (Requirement 11 and 16)

5.4 Maintenance of Data Provenance (Authenticity) and Integrity

- **Permanent Naming.** An archive needs a method of registering permanent names for the unique content in its collections. Because metadata may "reach inside" files, an archive needs a method of registering permanent names for data elements inside files, independent of possible permutative rearrangement of either the files or their format. (Requirements 8 and 9)
- **Provenance Tracking.** Data and metadata provenance are critical elements of an archive. Thus, the design of an archive's systems must include a process that updates the provenance of items within the archive after any archival operation and a process that verifies the provenance when needed. Given the volume of data and the high rate at which digital content can flow through an archive's systems, it is important to fully engage the computer portions of the archive in helping with this maintenance.
- **Transactional Basis for System Operations.** When the archive's systems transfer files or metadata form location to location, the archive needs to automate the process of ensuring that the transfer was authorized and completed satisfactorily. Transactions that can be rolled back if they are not successfully completed provide the assurance that the metadata and data will be consistent with one another.
- **Transaction Auditing and Reconciliation.** Just as an accountant reconciles the journals with the ledgers in a business setting, so the archive's systems must allow the record of transactions to be reconciled with the actual state of the archive's content inventory. Likewise, it is a requirement that the transaction accounts should be auditable – and that there be procedures for performing that work.

6. Mapping of Requirements onto Global Grid Forum standard groups

The category Level 1 requirements listed in Section 5 will impact the design of grid software. In this section we map the requirements to the GGF standards groups. We combine the category

Level 1 requirements with the original requirements to produce a set of criteria that the Grid software should meet:

Preservation requirements

- Preserve infrastructure independence
- Preserve authenticity
- Preserve integrity

Viability concerns:

- Minimize cost of ownership
- Enable high reliability

For each requirement, we outline impacts on the Global Grid Forum Standards Groups. The impacts will evolve based upon the standards created by each group. This will require iteration of requirements between the preservation environments community group and each standards group. For each group, the impact has been differentiated between utility of the proposed standard, preservation requirements, and viability concerns.

6.1 Infrastructure Standards Groups

The over-riding preservation requirement is how to ensure backward compatibility between the new versions of the standards and the prior versions of the standards. If a new standard requires a new protocol, then implementations are need that will be able to work with both the old and new protocols while the infrastructure is upgraded.

6.1.1 Ipv6

This group examines how to create network-neutral grid services. The Ipv6 standard addresses expansion of the network address space and provides additional network management functions. The document GFD.40 on “Guidelines for IP version independence in GGF specifications” provides essential information on how to build services that can handle the new Ipv6 addressing scheme. As such, this enables infrastructure independence across multiple versions of IP networks.

Utility:

- We expect the use of Ipv6 will be mandated in future preservation environments to ensure the ability to continue to be able to address remote storage systems over wide area networks.

Preservation Requirements:

- Compatibility of Ipv6 addressing with logical name spaces for resources. The IP address forms the physical address for each storage resource. The data grid logical resource name is the invariant address. Thus it will be possible to map to new forms of the IP address.

Viability:

- Protection against malicious users would be greatly aided by the ability to track the source of each network message. This means added functionality to tie all delivered messages to the origin IP address.

6.1.2 Network Measurement

This group is proposing standards for a common data model and format for reporting network measurements. They propose serializing measurement data in an XML schema, using standard attribute names to characterize the meaning of the data. This approach is compatible with the use of XML schema by the digital library community.

Utility:

- Multiple types of end-to-end measurements will be used by preservation environments to optimize use of networks.

Preservation Requirements:

- Of interest is the ability to map the Network Measurements schema to XML standards that may be required by the preservation community. An example is the METS Metadata Encoding Transmission Standard. The Network Measurements schema could be cast as a METS profile. It will then be possible to manage event information about network performance in the same structures that are used to manage preservation of material. An archive of event information could then be manipulated with existing digital library tools.
- A second interest is the ability to characterize the integrity of network transmissions when moving data within a preservation environment. The goal is to be able to make assertions about the reliability of the data transfers (from disk to disk) and the path over which the data was transmitted. Part of the authenticity information for submission to an archive should be the IP address from which the material was sent. This is related to tracking the chain of custody of a file. When a file is moved over a less reliable network, additional verification of the integrity of the file will be required. Network measurements need to include estimates of reliability of the transfer.

Viability:

- How will network measurements be incorporated in cost models?
- How can the measurements be linked directly to the archived material? Authenticity implies the need to associate each transfer with the logical file name, the logical name for the persons sending the data, and the logical names for the resources. These names are managed by the preservation environment independently of the remote site from which the data are transmitted.

6.1.3 Data Transport

This group promotes the creation of new standards for secure, robust, high-speed transport of data.

Utility:

- Preservation environments will need the secure, robust, high-speed transport of both data and metadata.

Preservation Requirements:

- Will the data transport mechanisms include the ability to aggregate metadata before transport? A similar capability is needed to pack small files before transport.
- How will end-to-end performance be assured? The transmission bottleneck may be the number of sources, the number of parallel I/O streams, or the number of receiving nodes?
- How will the security interact with preservation authenticity requirements, namely authentication against the logical name space that the preservation environment is using to identify individuals?

Viability:

- Preservation environments will need the ability to interoperate across multiple data transport protocols. Given that the congestion algorithms used by the transport mechanisms will probably be different across different versions, metrics are needed to evaluate the preferred transport mechanism for access to a given remote storage system. The metrics for evaluating which transport to use will in turn impact the standard schema developed by the network measurement group.
- A preservation environment will need to base its cost models on the expected data transport performance, and will need assessment tools to do the cost evaluation.

6.1.4 Grid High-Performance Networking

This group serves as a liaison to networking standards bodies.

Utility:

- As a liaison, it could promote interoperability standards between transport protocols. A reasonable goal is to minimize the impact of transport changes on preservation environment software as standards evolve.

Preservation requirements:

- For infrastructure independence, a preservation environment needs the ability to use any standard network protocol. As protocols evolve, an upgrade path is needed that minimizes the impact on preservation software.

Viability:

- Ease-of-use dictates that the choice of network be automated. What mechanisms will be provided to enable the preservation processes to select the appropriate network, and then use the resulting protocol? Will there be mechanisms to negotiate network protocol between services?

6.1.5 Network Measurement for Applications

This group focuses on the development of network-aware middleware, which is able to exploit knowledge about network parameters including bandwidth, latency, and jitter.

Utility:

- The ability to use the correct protocol is important for delivery of advanced data products such as video streams out of the archive.

Preservation Requirements:

- The implication is that the type of data product will influence the choice of transport mechanism. This in turn implies that the preservation environment will have to support multiple transport mechanisms, and that the choice will depend on multiple parameters:
  - Network performance
  - Size of data to move
  - Type of data to move

  A generic solution is desired that minimizes the number of transport protocols that must be maintained by the preservation environment.

Viability:

- The incorporation of network performance knowledge into preservation processes implies that characterization of the network will use standard attributes. Will these standard attributes be uniform across all transport protocols, or will the preservation process require analysis of a different set of attributes for each protocol that is used?

6.2 Data Standards Groups

Preservation environments employ a radically different perspective on management of state information compared to grid services. The preservation environment associates all state information directly with each preserved item. The definition of authenticity is that this bond between the state information and the preserved item is never lost. Grids organize state information in service specific catalogs. A major challenge is migrating from multiple independent service information catalogs into a preservation catalog.

Most of the Grid standards relate to data access rather than data management. The challenge of authenticity is the guarantee that provenance metadata will be correctly associated with each registered file, that operations on each file can be tracked to check chain of custody, and that the grid state information is appropriately updated after every operation. At the moment, the management of consistent state information is left up to the preservation application; hence the development of data grids, which manage the state information self-consistently. An environment that imposes consistent state information management is possible if the following constraints are implemented:

- Data grid ownership of material. Unless the data grid owns the files that are stored at a remote site, it will not be possible to track operations done on the files.
- Data grid virtualization of trust management. The data grid acts as the surrogate for the archivist to ensure that access controls established by the archivist will be followed no matter where in the data grid the material is stored.
- Data grid control of the name spaces used to identify storage resources, files, users, and metadata. This requirement makes it possible to provide persistent naming for use by the preservation environment, even when material is migrated to new storage solutions.
- Decoupling of the presentation interface protocols from the storage repository access protocols. This makes it possible to manage access mechanisms independently of the choice of storage repository. The expectation is that more cost-effective storage solutions will be found as technology evolves. Migrating to the new storage systems should no impact the access mechanisms.

These requirements enable complete data virtualization, the decoupling of the management of data from the storage systems in which they are deposited.

6.2.1 Data Access and Integration Services

This group promotes standards for consistent access to autonomously managed databases.

Utility:

- Preservation environments will also archive databases, and will need standard services for accessing archived databases.

Preservation Requirements:

- A standard access mechanism is needed for both archived and new databases. This requires the ability to write drivers to interact with new database technology, and the ability to maintain the drivers for interacting with prior database technologies. At the moment, separate access services are used for each type of database. A single access service is desired.
- Stable design parameters fare needed for the access mechanisms. As new database features are added (such as descriptions of relationships that must be true for a metadata attribute name to be meaningful), the interface should still support prior access mechanisms.
- Support for bulk operations on databases is needed. Since archives will manage hundreds of millions of files, the ability to register new metadata in bulk is needed.
- Mechanisms are needed to assert the validity of the metadata. This includes correspondence to authenticity information encapsulated in Archival Information Packages (AIPs), self-consistency checks on the metadata (presence of required attributes), and integrity checks that the metadata is not corrupted (checksums).
- Access controls on metadata. Roles are needed for creating and updating metadata along with authentication and authorization of use of the roles.

Viability:

- The preservation environment will manage the lifetime of the archive. The grid services will need to interact with the life-cycle management policies of the preservation environment.

6.2.2 Grid File Systems

This group promotes standards for describing and organizing file-based data.

Utility:

- The Grid File System Directory Service could manage the namespace of federated and virtualized data across file system resources.

Preservation Requirements:

- Support is needed for Archival Information Packages (AIP). An AIP is a mechanism to both aggregate authenticity metadata with the corresponding file, and a mechanism to aggregate multiple files before storage. The ability to reference both the authenticity metadata and each file is needed, even when the AIP is stored as a single package.
- Support is needed for authenticity. This implies the ability to track all operations, accesses, updates performed upon both files and the authenticity metadata. It is vital in an archive that the link between authenticity metadata and the corresponding files be preserved across all operations performed upon the files.
- Support is needed for chain of custody. The name space identifying archivists should be decoupled from the user name space managed by the storage system. The preservation environment should manage the distinguished user name space, and store data under its control on a single account within the storage system. This implies that access permissions are controlled above the level of the grid file system.
- Support is needed for integrity through the ability to create and validate checksums.
- Support is needed for execution of archival processes at the remote storage system, such as checksum validation.
- Support is needed for federation across independent grid file systems. This helps minimize risk of data loss due to operational procedures within a single data grid.

Viability:

- Support is needed for minimizing cost of ownership. A middleware implementation is preferred rather than an operating system kernel mod, as middleware is more easily ported across new types of storage systems.
- Support is needed for scalability. A critical element is the ability to query the directory service without actually accessing the storage system to retrieve information about the stored data.
- Support is needed for multiple types of data copies. At a minimum, the archivist workspace will need the ability to create replicas (synchronized copies), versions (numbered copies), backups (time-stamped copies), and independent copies of files.

6.2.3 Data Format Description Language

This group promotes the creation of an XML-based language to describe the structure of binary and character encoded files and data streams.

Utility:

- This project can create the standard representation for data encoding formats that allows the construction of generic parsing routines. This is turn makes it possible to decouple data parsing from applications that operate on the structures for display and manipulation. The hope is that a new application can define the set of operations that it will perform, map these operations to a standard set supported by the DFDL libraries, and then perform the operations on the structures that have been characterized by DFDL.

Preservation Requirements:

- A generic XML-based description is needed that can describe all types of data formats. Of interest are complex encodings such as databases, GIS systems, and office products.

Viability:

- The tools that process the encoding format need to be implemented in a portable language to ensure that the system will function on future operating systems.

6.2.4 GridFTP

This group promotes improvements to the GridFTP protocol for bulk file transfer, including parallel transfer, GSI authentication, and striped transfers.

Utility:

- Reliable data movement will be needed by preservation environments.

Preservation Requirements:

- Bulk transport of small files is needed. This requires packing small files before movement using parallel I/O streams.
- Bulk registration of files is needed. This requires packing metadata about each file (size, name, time stamps), moving the metadata in bulk, and loading into a metadata catalog. An example is the recursive registration of an existing directory into a preservation environment.
- Support for execution of remote procedures. An example is metadata extraction, the packing of the metadata into an XML file, the movement of the file, and the registration of the metadata into the preservation environment.

Viability:

- The system needs to operate on the name spaces managed by the preservation environment.

6.2.5 Grid Storage Management

This group promotes a standard Storage Resource Manager to support dynamic space allocation and file management of shared storage components on the Grid. Capabilities include storage reservation and information on storage availability.

Utility:

- The management of storage resources, including the staging of files and space reservation, will be needed by preservation environments.

Preservation Requirements:

- The system should support integrity functions including checksum validation.
- The system should work with the preservation environment logical name spaces.

Viability:

- The system should work off of the same metadata catalogs used by the preservation environment.
- The system should be portable onto new operating systems.

6.2.6 Information Dissemination

This group is defining the low-level operations needed to support data and event dissemination, and a high-level interface for information dissemination.

Utility:

- The management of preservation processes requires the ability to detect when failures occur, and could be based on event dissemination information.

Preservation Requirements:

- The integration of event notification with workflow systems can improve the ability to track results of preservation processes.

- Grid services support the virtualization of workflows. The ability to maintain information about workflows is essential for tracking the preservation processes that have been applied.

Viability:

- A true workflow virtualization system is needed for workflow processes. For the grid services to be useful, they need to be integrated into a coherent system that tracks the results of the application of the services and associates the processing events with each processed data object.
- Of interest is interaction with other metadata transport mechanisms, such as the Open Archives Initiative, Protocol for Metadata Harvesting - OAI-PMH.

6.2.7 OGSA Data Replication Services

This group is refining grid service specifications for data replication services, in particular catalogs about data location.

Utility:

- Preservation environments need to manage multiple replicas to mitigate risk of data loss.

Preservation Requirements:

- The replicas need to use the logical name spaces managed by the preservation environment.
- Multiple types of copies of data are needed, including transformative migrations to alternate encoding formats, backups (time-based snapshots), versions (numbered copies). State information is needed to differentiate between these types of copies.
- Support is needed for replication of AIPs.
- Support is needed for replication between independent data grids.
- Support is needed to ensure that authenticity and integrity metadata remain linked to replicas.

Viability:

- Preservation environments associate state information with each preserved item. A mapping will be needed between the replica state information catalog and the preservation metadata catalog.

6.2.8 Transaction Management

This group is applying transaction management techniques to grid systems for updates.

Utility:

- A key component of preservation is tracking chain of custody. Transaction management would enable the ability to ensure that each preservation process completes, or is rolled back.

Preservation Requirements:

- Transaction management needs to be integrated with workflow environments, with resulting state information saved for each item after the application of each workflow process.

Viability:

- The resulting system needs to support bulk operations on collections of files and metadata. The challenge is how to apply transaction processing to groups of files. One approach is to build re-entrant processes, such that the process can be re-executed to correct partial application. The other approach is to roll back the state of the entire submitted group of files and re-try from scratch.

6.2.9 OGSA Data

This group is designing an overall architecture for virtualization of workflows. This includes the message patterns and interfaces for integrating grid services.

Utility:

- The management of state information for workflows is necessary to manage chain of custody during application of preservation processes.

Preservation Requirements:

- Authenticity requires the tracking of all operations done on data.
- Integrity requires automation of validation mechanisms for checksums

Viability:

- The workflows will need to manipulate collections of data, manage error conditions, and maintain consistent state information about operations performed upon data.

6.2.10 Byte IO

This group is designing an interface to read sequences of bytes from multiple types of resources.

Utility:

- Preservation environments need the ability to read data from multiple types of storage systems.

Preservation Requirements:

- Current data grids provide support for POSIX byte I/O.

Viability:

- Additional operations are used to support metadata extraction, bulk operations on remote data, aggregation of data. Preservation environments use more than simple byte I/O in remote operations.

6.3 Compute Standards Groups

Preservation environments apply preservation processes (appraisal, accession, arrangement, description, preservation, and access) when ingesting data. These processes can run in a grid environment. Hence there is strong interest in the ability to apply the preservation processes at the locations where there is sufficient compute power and storage for each record series.

6.3.1 Grid Resource Allocation Agreement Protocol

This group is producing a common resource management protocol for advanced reservation of resources.

Utility:

- Distributed processing of workflows will become important as the volume of material to be archived increases.

Preservation Requirements:

- Authenticity requires the ability to track the location where preservation processes are executed.

Viability:

- Standard grid services should be sufficient for allocation of grid resources. What is not obvious, is whether allocation mechanisms will be provided for multiple types of storage systems (disk caches, on-line collections, archives).

6.3.2 Job Submission Description Language

This group is specifying an abstract Job Submission Description Language for interacting with popular batch systems.

Utility:

- Interaction with batch systems will be part of large scale processing demands for preservation environments.

6.3.3 Grid Scheduling Architecture

This group is defining a scheduling architecture that can control use of networks, software, storage, and processing units, and the interactions of these systems with data management.

Utility:

- Again scheduling of resources will be important for processing large amounts of material.

6.3.4 OGSA Basic Execution Services

This group is designing a set of Execution Management Services. This is equivalent to the creation of workflow management.

Utility:

- The management of processes executing in a distributed environment is needed for large scale processing.

Preservation Requirements:

- Authenticity requires the tracking of processes applied to each file, and associating the process execution state with each file.

Viability:

- Consistent management of state information resulting from application of grid services is essential in preservation environments.

6.4 Architecture Standards Groups

The current architecture is being devised independently of the standards used for data management. An integration of the architecture used for grid services and the architecture used for digital libraries is needed. This requires a change in perspective towards identification of the name spaces needed to manage data rather than execute services. Integration with data management standards is also needed:

- METS Metadata Encoding Transmission Standard
- Archival Information Package
- Open Archives Initiative Protocol for Metadata Harvesting
- Authenticity and integrity metadata

6.4.1 Open Grid Services Architecture

This group is developing an architecture roadmap for grid services, focused on functionality and interrelationships between OGSA services. A major challenge is support for bulk operations, in which grid services are applied to a collection of files. A desire is the use of re-entrant services, such that partial completion of the processing of a collection can be completed by the re-application of the service.

6.4.2 Grid Protocol Architecture

This group is developing a conceptual framework for grid services that focuses on a minimal set of protocols. Preservation environments need the simplest possible implementation to improve sustainability and robustness.

At the same time, preservation environments want to be able to archive the information content that has been created by a service. This corresponds to checkpointing the service, and reliably storing the associated information content. A preservation environment will then want to re-instantiate the service, possibly bringing up the service on a new hardware infrastructure. The ability to checkpoint services needs to be designed into the grid services environment.

6.4.3 OGSA Naming

This group is implementing a three-level naming specification for web services (WS-naming, WS-addressing, physical address). The goal is to support interoperability between different name resolution services. Current preservation environments rely on a two-level naming specification (logical name, physical address). A second issue is that preservation environments need two-level naming for files and resources, while maintaining control of name spaces for users and metadata.

6.5 Applications Standards Groups

The development of higher-level services requires simplified interfaces to grid services. Requiring a larger number of levels of software to implement applications will lead to an environment that is harder to maintain.

6.5.1 Grid Remote Procedure Call

This group is defining a grid remote procedure call. Most grid services will benefit from the use of remote procedure call style invocation of remote operations. This minimizes the number of messages that need to be sent over wide-area-networks. This is particularly important for manipulating large numbers of small files.

6.5.2 Grid Information Retrieval

This group is defining an architecture for information retrieval, including document collection management, indexing, searching, and query processing. The integration of document collection management with the technologies coming from the digital library community is needed to ensure compatibility. In particular, the two communities are pursuing different standards for organizing information (METS) and for managing information (AIPs).

6.5.3 Distributed Resource management Application API

This group is developing an API specification for controlling jobs submitted to Distributed Resource Management Systems. This includes submitting, terminating, and suspension. These capabilities will be needed for large scale processing.

6.5.4 Simple API for Grid Applications

This group is defining a simple API for remote job submission, file transfer from within a program. The simple API may not provide the set of bulk operations needed to manipulate large numbers of small files.

6.5.5 Grid Checkpoint Recovery

This group is defining a user-level API and associated services to permit check-pointed jobs to be recovered and continued. This will be needed for large-scale processing of data.

6.6 Management Standards Groups

The management of policies is a critical component of a preservation environment. When to apply a preservation policy such as migration to new technology, media refresh, transformative migration, checking of Service Level Agreements for storage, producer-archive submission agreement validation, is needed as a general capability.

6.6.1 Application Contents Service

This group is defining central management for descriptions of applications. They are developing an Application Repository Interface, and an Application Archive Format for bundling components of an application. Preservation environments will need to apply this to both grid services and preservation processes, while asserting consistency requirements.

6.6.2 Configuration Description, Deployment, and Lifecycle Management

This group is developing mechanisms to describe configuration of services, their deployment, and management of their deployment life cycle (instantiation, initiation, start, stop, restart). Note that data management requires long-running services (the catalog that manages a collection will never by turned off). However the ability to recreate a service is a critical component of the management of technology evolution. Consider the following:

- Create an information management service
- Checkpoint the information management service. This means that all information required to implement the service is archived, along with all of the information content being managed by the service.
- Archive the checkpoint.
- Retrieve the checkpoint
- Re-instantiate the service on new technology, demonstrating that no information content has been lost

The result is the migration of the long-running service onto new technology.

6.6.3 Grid Economic Services Architecture

This group is defining protocols and service interfaces to support multiple economic models for the charging of Grid services. This includes the provision and consumption of grid services, a grid banking service for financial transactions, and a chargeable Grid service to encapsulate existing services and charge for use.

6.6.4 OGSA Resource Usage Service

This group is defining a Resource Usage Service to track resource utilization. This is a component that is required for building cost models for the preservation environment.

6.6.5 Usage Record

This group is defining a common usage record for interchanging accounting and usage information. This component will be needed to support large-scale processing across multiple resources.

6.7 Security Standards Groups

The preservation of integrity requires the ability to control access for privileged write operations, for updating metadata, for accessioning new material.

6.7.1 Open Grid Service Architecture Authorization

This group is defining specifications for interoperability and pluggability of authorization components. They plan to leverage SAML and XACML. This will be needed for integrating the Shibboleth approach to trust used in digital libraries with the Grid Security Infrastructure.

6.7.2 OGSA-P2P-Security

This group is defining how to build grids that access desktop systems to enable distributed computing. The security requirements must handle the situation where the machine user is also the administrator. One approach is to integrate data virtualization with workflow virtualization through installation of virtual Machine environments across the desktop systems. The ability to guarantee that processing has not been compromised and that the resulting data products remain under archivist control is essential before desktop systems will be used for building preservation environments.

6.7.3 Firewall Issues

This group is examining data transport policy enforcement. This is the enforcement of policy decisions on behalf of participating systems used by an application. Network examples include firewalls, network address translators, application level gateways, and VPN style gateways. This technology will be needed for producer-archive submission pipelines that may go across firewalls.

6.7.4 Trusted Computing

This group is evaluating how trusted computing initiative concepts can be applied in grids. This includes using hardware modules to enable and manage data and individual identities. The use of hardware modules may be the ultimate mechanism for establishing trust within a preservation environment.

7. Summary

The requirements that must be supported by a digital archive can be cast as constraints on both the choice of system architecture and the management policies required to maintain the digital archive. An examination of the experiences of the NASA Langley Research Center result in a set of sixteen recommendations for digital archive requirements. The sixteen recommendations can be mapped to research groups within the Global Grid Forum as requirements on the grid software infrastructure.

8. Author Information

Bruce Barkstrom
Atmospheric Science Data Center
NASA Langley Research Center
b.r.barkstrom@larc.nasa.gov.

9. Acknowledgements

The ideas expressed here were developed under projects with the National Aeronautics and Space Administration.

10. Glossary

The terms used to describe digital archives are listed in this section.

Archival engineer – the system administrator of a digital archive.
Chain of custody – the organizations that maintain the archive, the storage systems used to hold the data, and the processes that have been applied to the data while in the archive.
Context – the ancillary information needed to understand the relevance of data, including the ability to interpret, use, and apply the data in research.
Digital archive – the software and hardware systems used to manage data for periods of time exceeding the lifetime of any single software or hardware component.
Exception handling – the application of automated processes to identify problems and manage the response to the problems, while notifying the archival engineer.
Intellectual capital – the standard digital reference data sets that are used to support research within a scientific community, along with the metadata that
Open Source – software systems for which the source code is distributed, and for which user-specified modifications can be incorporated independently of the distributor.
Permanent names – persistent identifiers for data, archivists, metadata, access constraints, and storage resources that remain invariant when new technology or media are incorporated in the archive.
Provenance – the set of metadata describing the origin of the data, and the calibration files and calibration programs used to generate a derived data product or the simulation code used to generate simulation output.
Technological obsolescence – the replacement of software or hardware components by new technology that is more cost effective.

11. Intellectual Property Statement

The GGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the GGF Secretariat.

The GGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the GGF Executive Director.

12. Full Copyright Notice

13. References

1. InterPares Preservation Task Force, “How to Preserve Authentic Records”, Oct. 2001, http://www.interpares.org/book/interpares_book_o_app06.pdf
2. Moore, R., A. Rajasekar, “Common Consistency Requirements for Data Grids, Digital Libraries, and Persistent Archives”, Grid Procotol Architecture Research Group draft, Global Grid Forum, April 2003
3. Moore, R., “The San Diego Project: Persistent Objects”, Proceedings of the Workshop on XML as a Preservation Language, Urbino, Italy, October 2002.
4. Moore, R., A. Merzky, “Persistent Archive Concepts,” Global Grid Forum, December 2003.
5. Moore, R. (2000a), “Knowledge-based Persistent Archives,” Proceedings of La Conservazione Dei Documenti Informatici Aspetti Organizzativi E Tecnici, in Rome, Italy, October, 2000.
6. OAIS - Reference Model for an Open Archival Information System (OAIS). submitted as ISO draft, http://www.ccsds.org/documents/pdf/CCSDS-650.0-R-1.pdf, 1999.
7. Thibodeau, K., “Building the Archives of the Future: Advances in Preserving Electronic Records at the National Archives and Records Administration”, U.S. National Archives and Records Administration, http://www.dlib.org/dlib/february01/thibodeau/02thibodeau.html
8. Underwood, W. E., “The InterPARES Preservation Model: A Framework for the Long-Term Preservation of Authentic Electronic Records”. Choices and Strategies for Preservation of the

Collective Memory, Toblach/Dobbiaco Italy 25-29 June 2002. To be published in Archivi per la Storia.

Areas Contributing to Design Requirements

Core Requirements

Table 1. Automated Generation of Search Interface

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	Hand Development of Web Pages for Search Interface	Below
2	Single Page Template with Static Web Page Generation	Below
3	2 + Multi-Type Page Templates with Automated Static Web Page Generation	Below
4	3 + Active/Interactive Web Pages (XUL, CSS, Jscript)	Below
5	4 + Alternate Technologies (DHTL, CSS, Jscript) with On-the-Fly Page Generation	Above
6	5 + User Controllable Attribute Visibility	Above

Table 2: Cost Effective Operations

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	Ad Hoc, Manual Record Keeping Sufficient for Time and Attendance; Informal Planning	Below
2	Periodic, Hierarchical Planning; Automated Record Keeping	Below
3	Periodic, Hierarchical Planning Process, using Deterministic Cost Model for Forecasting	Above
4	Periodic, Hierarchical Planning Process, using Stochastic Schedule and Resource Management	Above

Table 3. Automated Trouble Ticket Process

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	Manual, Ad Hoc Process for Fault Detection, Diagnosis, and Correction	Below
2	Written, Manual Procedures for Fault Detection, Diagnosis, and Correction	Below
3	Automated Fault Detection with Written Manual Procedures for _Fault Diagnosis and Correction	Below
4	Automated Fault Detection and Diagnosis with Written Procedures for _Fault Correction	Below
5	Automated Fault Detection and Diagnosis with Automated Fault Correction _or By-Pass; On-Call Help	Above

Note: The planning process should also evaluate the cost-effectiveness of new Technology

Table 4. Automated System Installation

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	Manual Installation of All Components and Content	Below
2	Manual Installation of Base Software (e.g. Compilers, Scripting Languages, Databases),_Archive Infrastructure, and Scripted Installation of Archive Contents	Below
3	Manual Installation of Archive Infrastructure; Automated Installation of Base Software and_of Archive Contents	Below
4	Automated Installation of All Components and Content (Necessary for Automated _Archive Replication)	Above

Table 5. Commodity Computers and Data Storage

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	Custom-Designed Components with Proprietary Software	Below
2	Commodity Computers and Data Storage Elements with Open Source Software	Above

Note: The search interface may use a reserved vocabulary that is derived from provenance metadata or a thesaurus. The use of Open Source software may require that the archival engineers maintain, build, and validate the software used within the archive.

Table 6. Open Source Software to Reduce Licensing Costs

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	System Built from Proprietary Software	Below
2	Less than Half of System Components (by cost) from Proprietary Software	Below
3	More than 90% of System Components (by cost) from Open Source Software	Above

Note: The management costs for maintaining an appropriate version of the Open Source software for the local archive should be compared with commercially supported software, or amortized through an archival consortium.

Table 7. Automated Hardware and Software Inventory and Configuration Management

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	Manual, Ad-Hoc Processes	Below
2	Hardware and Software Inventory in Database; Manual Configuration Management	Below
3	Hardware and Software Inventory and System Configuration in Database _with Manual Updates	Below
4	Hardware and Software Inventory in Database with Automated Checks of Inventory;_System Configuration Automatically Maintained	Above

Table 8. Migration

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	System Backed Up to Temporary Storage; Ad Hoc Procedures	Below
2	On-Site Backup Only; Data and System Transfer Done with Ad Hoc Procedures	Below
3	Off-Site Backup, Automated Data and System Transfer with Verification	Below
4	3 + Deterministic Technology Migration Model _(e.g. “Moore's Law” and related “rules of thumb”) to Plan Future Migration Steps	Above
5	3 + Stochastic Technology Migration Model and _Options-Based Pricing of Investments for Future Migration Steps	Above

Note: The automation of the network configuration is equally important, especially when managing data that has been replicated to another site.

Table 9. System Disaster Recovery

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	Archive Reconstruction	Below
2	1 + Valuation and Risk Analysis	Below
3	2 + Off-Site Backup with Weekly Deliveries of New Archive Contents	Below
4	3 + Periodic Disaster Rehearsals	Below
5	4 + Inventory and System Reconstruction Checks on a Systematic Basis	Above
6	5 + Multi-Site Replication with Voting and Verified Formal Model of Disaster Probabilities	Above

Table 10. Automatic Diagnostics

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	System Error Messages	Below
2	Unified Error and Exception Handling System Tied to Use Cases	Below
3	2 + Automated Fault Detection Log with Robust Exception Handliing	Above

Note: The ability to rebuild the name spaces used to identify data, archivists, resources, provenance metadata, and access controls across the multiple sites is essential for minimizing risk of data loss.

Table 11. Designed-in Security

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	System Provides Passive Protection Behind Firewall	Below
2	1 + System Actively Monitors Most Interactions with External Entiies	Below
3	2 + System Actively Monitors Internal Data and Components	Below
4	3 + System Actively Monitors and Reconciles All Internal Transactions	Above

Table 12. Multi-Language Implementation

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	Core Archive Components Implemented in One Language	Below
2	Core Archive Components Implemented in Two Languages	Above
3	Core Archive Components Implemented in Three Langauges	Above
4	Core Archive Components Implemented in Two or More Languages _and Maintained by Open Source Community	Above
5	Open Source and Proprietary Systems Available for Highly Robust Archive	Above

Note: The management of security risk due to the compromise of system administrator accounts implies the need for a deep archive that will not be accessible by users.

Table 13. Use Cases Become System Manuals

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	No Use Cases	Below
2	Use Cases Developed to Identify System Objects	Below
3	2 + Design and User Manuals Based on Use Cases	Below
4	3 + Test Procedures Based on Use Cases	Above

Table 14. Systematic Procedure for Updating and Validating Use Case Evolution

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	No Use Cases	Below
2	Use Cases Developed, But Not Updated	Below
3	Periodic Review of Use Cases, with Updates for Documentation and Test Cases	Above

Table 15. Object-Oriented Design Traceable to Use Cases

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	Design Decoupled from Use Cases (**Unacceptable Design Practice**)	Below
2	Original Design Derived from Use Cases	Below
3	2 + Substantial Portion of Test Procedures Derived from Use Cases	Below
4	3 + Design Coevolves with Use Cases	Above

Table 16. Formalization of Message-Passing Protocols

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	No Systematic Organization of Message-Passing Protototocols (**Unacceptable Design Practice**)	Below
2	Internal and External Protocols Documented from Objects Derived from Design Based on Use Cases	Below
3	2 + Protocols Coevolve with Use Cases and Object Design	Above

Table 17. Systematic Procedure for Documenting Protocol Evolution

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	No Documented Protocols	Below
2	Protocol Documentation Not Updated after Initial Design	Below
3	Protocols Periodically Reviewed and Systematically Updated	Above

Note: Protocols that ensure backwards compatibility minimize risk of data loss when migrating an archive to software systems using a new protocol.

Table 18. Formalization of Operational Procedures (CMM)

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	Undocumented, Ad Hoc Operational Procedures	Below
2	Documented Procedures, with Some Ad Hoc Deviations	Below
3	Documented Procedures Periodically Reviewed and Updated with Statistical Data	Above
4	Documented Procedures Based on Formal Model of Effective Organizational Communication Patterns	Above

Table 19. Training

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	Ad Hoc	Below
2	Training Necessary for Understanding and Using Base Components of System _(e.g. OS, Compilers, Scripting)	Below
3	2 + Training Necessary for Understanding and Using Infrastructure Components of_System (e.g. Objects internal to Archive)	Below
4	3 + Training Necessary for Understanding and Modifying Archive Contents	Below
5	4 + Training Necessaray for Evolving the Archive	Above

Table 20. Procedures for Operational Procedure Evolution

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	Undocumented Procedures (**Bad Operational Practice**)	Below
2	Operational Procedures Documented	Below
3	2 + Systematic and Periodic Review	Above

Table 21. Peer Review of Design and Documentation

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	No Documentation	Below
2	Peer Review within Local Developer Community	Above
3	2 + On-Site, Authorized User Peer Review	Above
4	3 + Full User Patricipation in All Development Activities	Above

Note: Peer Review is generally regarded as highly beneficial to producing system components of high quality. Peers are members of the development team or individuals respected by the development team for their knowledge of the systems and their skills in solving problems. Peer Reviewers must take the time to understand and review the content of the material being reviewed. They may also suggest solutions to problems that are discovered in the course of the review.

An important intent of peer review is to make the design, code, and procedures part of the development community's knowledge, rather than being the property of a single individual.

Peer Review is not equivalent to a Process that engages in large, Formal Reviews, in which the development team invites large numbers of people to watch specially prepared review packages. The Airlie Software Council has identified Formal Reviews as one of the nine worst software development practices. In the experience of the authors, Formal Reviews are political in nature and waste extraordinary amounts of time and energy. They also exhaust the developers and waste large amounts of paper and other scarce resources.

Table 22. Open Source Publication of Design and Code

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	No Publication	Below
2	Publication of Design and Code	Above

Table 23. Ability to Interact with Open Source Community and Incorporate Open Source Contributions

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	No Open Source Community Involvement	Below
2	Publication of Design and Source Code in Open Source Form	Above
3	2 + Active Solicitation of Contributions from Open Source Community	Above
4	3 + Formal Organization of Open Source Project	Above

Table 24. Ability to Cooperate in Archive Federations that Maintain Local Autonomy

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	No Participation in Archive Federations	None given
2	Voluntary Participation in Federations with Goal of Interoperability	None given
3	2 + Resource Sharing Between Federation Members	None given

Note: There have been a number of studies and simulations of communities that engage in resource sharing. These suggest that community members may participate in resource sharing without the need for centralized oversight. We recognize the variability in the needs of various kinds of archives, which may allow some archives to operate in a solitary fashion, while others benefit greatly from the participation in federations. Accordingly, we do not set a minimum level of cooperation in order to meet Archive Core Requirements.

Table 25. Intellectual Property Rights Considerations Included in Design

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	No Recognition of Intellectual Property Rights in Use Cases or Design	Below
2	Intellectual Property Rights Controlled by Access Control Lists	Below
3	2 + Allowance for Internal System Partitioning of IPR	Above
4	3 + Procedures Developed to Allow Narrowing or Broadening of IPRs	Above

Table 26. Permanent File and File Content Naming and Registration

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	File Names and File Content Names Assigned without Consideration of Permanence	Below
2	Files Have Permanently Registerable Identifiers as well as names	Above
3	2 + File Contents have Structural Indexes that Allows Permanent Reference to and _Retrieval of Individual Data Elements – even in the event of Data Migrations and_Transformations	Above

Note: This set of requirements arises from the need to be able to uniquely identify files and their contents. We envision that the contents of the files must also be permanently identifiable, even if the data must migrate from one storage medium to another or be reformatted as a result of software or hardware obsolescence.

A simple example of the kind of capability needed for file content naming is the identification of the pixels in a satellite image that have been identified by a data mining algorithm as belonging to a particular hurricane instance. It is quite possible that the file would be recognized during data migration, so that what my have been included in the original data have been reorganized by data reformatting into a very different format or even broken into separate files.

Implementation of this capability will almost certainly require creation of Information Packages (in the sense defined by the OAIS Reference Model) that can refer to format transformations endured by the data after its original archival.

Table 27. Verifiable System for Maintaining Chain-of-Custody

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	No Verifiable System Incorporated into Design for Maintaining Chain-of-Custody of Archive Contents	Below
2	System Design incorporates systematic procedure for recording data transfers based on the Negotiated Submission Agreement between a Data Provider and an Archive – although the Archive maintains no automated procedure for verifying the transfers from the Data Provider or within the Archive.	Below
3	2 + Automatic Recording of Data Ingest and Transfer	Below
4	3 + Transaction Based Data Ingest and Transfer Operations	Below
5	4 + Auditing, Reconciliation, and Periodic Inventory of Archive Contents and User Accesses	Above

Table 28. Access Control and Authentication

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	No Access Control or Authentication Procedures	Below
2	Access Control Lists Maintained with Manual Procedures	Below
3	Access Control Lists Maintained with Automated Procedures and Transactional Auditing and Reconciliation	Above
4	3 + Authentication of Users (both Internal and External)	Above
5	4 + Physical Isolation of Archive	Above

Table 29. Transactional Basis for System Operation

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	No Record of System Messages Between Objects	Below
2	Messaages Generate Logs used with Manual Monitoring	Below
3	All System Activities Operate as Transactions that can be rolled back	Below
4	3 + Automated Fall-back and Recovery Meanchisms	Above

Table 30. Transaction Auditing and Reconciliation

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	No Transactions	Below
2	Messages Recorded in Logs with Manual Monitoring	Below
3	All activities operate on a transactional basis with automated roll-back, backup, and restore capability	Above
4	3 + Transaction and Reconciliation used to develop statistical data used for monitoring the system reliability in data handling operations	Above

Engineering Requirements

Table 31. Logical Name Space Complexit

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	Flat, One Layer	Not Specified
2	Multi-Layer, Homogeneous	Not Specified
3	Multi-Layer, Inhomogeneous Nodes	Not Specifed

Table 32. Metadata Complexity

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	Only Permanent ID and Full File Path Name (Logical Name) in Flat (One-Level) Hierarchy	Not Specified
2	Multi-Level Logical Name Space with Permanent ID and Full File Path Name	Not Specified
3	Multi-Level Logical Name Space with Single Table (no one-to-many or many-to-many relations) of Attributes at each LNS node	Not Specified
4	3 + Junction Tables (one-to-many and many-to-many relations) of Attributes at each LNS node	Not Specified
5	4 + Multiple Path Search Nodes	Not Specified
6	5 + Non-Homogeneous Logical Name Space Nodes	Not Specified

Probable User Success Requirements

Table 33. Search Interface Complexity

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	One File per Search – Permanent ID and Full File Path Name Searches Only (One Level) Hierarchy	Not Specified
2	Multiple Files per Search – Permanaent ID and Full File Path Name Searches Only	Not Specified
3	1 + Simple Logical Name Space Traversal (Multi-Click Navigation)	Not Specified
4	3 + Visibility of Node Metadata as a Table	Not Specified
5	4 + Multiple Files Selectable per Search	Not Specified
6	5 + SQL Query using Visible Buttons in Web Pages	Not Specified
7	6 + Multi-Entry Concept Maps	Not Specified
8	7 + Multi-Persona Searches	Not Specified

Table 34. User Help Complexity

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	User Manuals Only	Not Specified
2	1 + User Help Desk with Phone and Email Responses	Not Specified
3	2 + Interface Input Error Messages	Not Specified
4	3 + On-Line Tutorials in Context	Not Specified

Table 35. Data Distribution Complexity

Level	*Complexity Description*	*Minimum Level to Meet_Archive Core Requirement*
1	FTP Pull	Not Specified
2	1 + FTP Push	Not Specified
3	2 + Single Kind of Media Distribution	Not Specified
4	3 + Multiple Kinds of Media Distribution	Not Specified
5	2 + Specialized Subsetting and Reformatting Web Services	Not Specified
6	5 + Specialized Transformational Web Services (e.g. Visualization)	Not Specified